

Esame di Laboratorio di Fisica Computazionale

13 luglio 2018, ore 14.00

1 Mathematica

1. Si consideri la circonferenza Γ di raggio $R = 3$ centrata nel punto $C = (2, 3)$ e la corda che questa stacca sulla retta di equazione $y = -\frac{1}{2}x + 3$.
 - Si rappresentino in un solo grafico la circonferenza e la retta.
 - Si calcolino le coordinate dei punti di intersezione A e B tra la circonferenza e la retta.
 - Si calcoli l'area del triangolo ABC e l'area del settore circolare ABC .
Per ottenere questo risultato, si definisca una funzione che accetta come argomenti due vettori bidimensionali (o alternativamente le componenti) e che restituisce il prodotto scalare tra questi due vettori.
 - Si rappresentino, nello stesso grafico, oltre alla retta e alla circonferenza, anche i segmenti AC e BC .
2. Si risolva l'equazione differenziale

$$y''(t) - a_f \cos(\omega_F t) + \omega_H^2 y(t) = 0, \quad y'(0) = 0, \quad y(0) = 0 \quad (1)$$

- Si valuti la soluzione quando $\omega_F = \omega_H$.
- Si definisca una funzione dei parametri $(a_f, \omega_F, \omega_H)$ e della variabile t come la soluzione dell'equazione, in grado di rappresentare tutte le possibili scelte dei parametri.
- Posto $a_f = 1$, si rappresentino in un solo grafico, con $t \in [0, 200]$ i tre casi:
1) $\omega_H = 1, \omega_F = 0.1$, 2) $\omega_H = 1, \omega_F = 1$, 1) $\omega_H = 1, \omega_F = 10$.

2 C++: Esercizio

Si risolva l'esercizio proposto, possibilmente includendo tutto in un unico sorgente. La sufficienza è raggiunta risolvendo correttamente i primi quattro punti.

Esercizio

1. Si scriva una classe `Config` che descriverà sequenze di 12 valori booleani (`true` o `false`). Si ponga come membro privato una variabile di tipo `std::array` che conterrà la sequenza. Si scriva un opportuno costruttore di default, che inizializzi la sequenza con valori tutti `false`. [Nota: `std::array` è un container sequenziale della STL con taglia fissata, in particolare `std::array<T,n>` contiene n oggetti di tipo `T`.]
2. Si scriva il costruttore di copie (con la sintassi della lista di inizializzazione) e una funzione pubblica `print` che stampi la sequenza su `cout` come una riga di valori 0 o 1.
3. Si scriva inoltre una *access function* che restituisca *per valore* l'array privato. Rispondere brevemente (in un commento): dall'esterno della classe `Config` è possibile usare questa funzione per modificare la sequenza? Se sì dire come; altrimenti dire come modificare il codice in modo che diventi possibile.
4. Si scriva una funzione membro `incr`, che modifichi la sequenza nel seguente modo. A partire dal primo elemento,
 - (i) se l'elemento è falso lo rende vero e si ferma
 - (ii) se invece è vero lo rende falso e riparte dal punto (i) considerando l'elemento successivo(si ferma quando trova il primo elemento falso o quando termina gli elementi; essenzialmente questa funzione implementa l'incremento in rappresentazione binaria). Si faccia in modo che `incr` sia concatenabile, nel senso che la sintassi `c.incr().incr()` incrementi due volte l'oggetto `c`.
5. Si scriva una funzione pubblica `energy`, che restituisca l'energia della sequenza, cioè un numero reale calcolato come il valore assoluto della differenza tra il numero di elementi `false` e il numero di elementi `true`. Si vuole che questa funzione abbia un comportamento polimorfico.
6. Si definisca una classe `Ising`, che erediti pubblicamente da `Config`. Questo tipo di sequenza avrà un'energia che dipenderà da un parametro reale g (lo si metta tra i membri privati). In particolare, l'energia sarà pari a $c_0 - gn$, dove c_0 è l'energia calcolata nella classe base e n è il numero di interfacce nella sequenza, cioè il numero di volte in cui un valore `false` e un valore `true` si trovano adiacenti. Si scriva un opportuno costruttore che prenda come argomento il valore di g , con *valore di default* 1. Si scriva l'*overriding*

della funzione `energy`, riutilizzando (tramite chiamata a funzione) il codice già scritto per la classe base.

7. Si scriva una funzione globale `csort`, che ordinerà in senso crescente gli elementi di un `std::vector` (passato come argomento) contenente oggetti di tipo `Ising`. La relazione di ordinamento è quella data dall'energia: una sequenza s_1 precede una sequenza s_2 se l'energia di s_1 è minore di quella di s_2 . Si usi l'algoritmo `std::sort(it1, it2, pred)`. Per fare questo sarà necessario definire un opportuno predicato `pred`.
8. Nel `main` si riempia un `std::vector<Ising>` di 4096 oggetti di tipo `Ising` (con parametro g positivo fissato), partendo dalla sequenza di valori tutti `false` e operando dopo ogni inserimento con la funzione `incr`, in modo da avere nel vettore tutte le possibili sequenze di 12 valori. Poi si ordini il vettore con `csort` e si controlli che la prima sequenza (cioè quella a minima energia) sia costituita da valori alternati.
9. Si scriva una versione *generica* della funzione `csort`, che possa funzionare anche con `std::vector` contenenti oggetti diversi da `Ising`. (Si dovrà rendere generico anche il predicato `pred`.) Si risponda in un commento: qual è l'unica funzionalità che è richiesta agli oggetti contenuti in tale container perché il codice compili? Bonus: si renda il codice ancora più generico, in modo che `csort` funzioni anche con container diversi da `std::vector`.
10. Se si volesse istanziare e usare un `std::set` di oggetti di tipo `Config` si dovrebbe specificare un criterio di comparazione tra questi ultimi. Si vuole che tale criterio sia proprio quello implementato dal predicato `pred`. Aggiungere al codice l'opportuna definizione di `Comparator`, in modo che il seguente `main` funzioni:

```
std::set<Config, Comparator> s;  
Config conf;  
s.insert(conf);
```

3 Python

Il comando Unix:

```
find -size +1M -ls > all_big_files.txt
```

genera il file di testo "all_big_files.txt" in cui ad ogni riga sono contenute varie informazioni su ogni file che nella propria \$HOME supera la taglia di 1 Mbyte. Si scriva uno script in Python che esegua questo comando e che dal file all_big_files.txt estragga solo le linee relative a tutti i files che terminano col suffisso .txt e che infine scriva queste linee complete in un file di testo il cui nome passato direttamente da riga di comando.